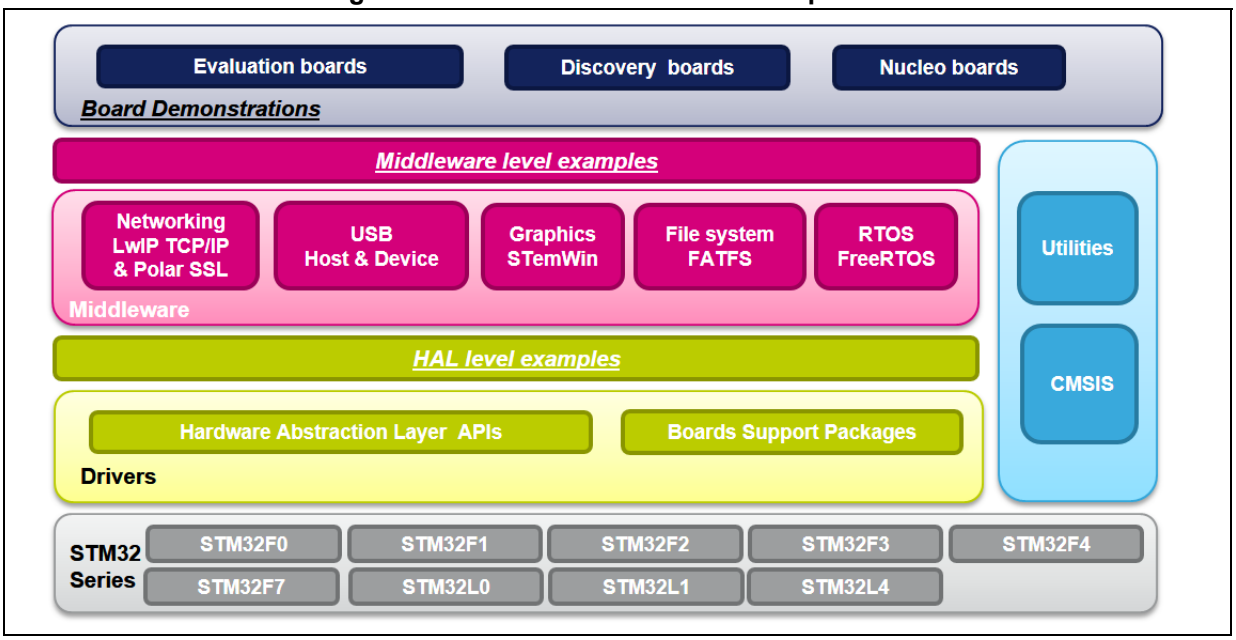
AN4731
Application note

STM32Cube firmware examples for STM32F7 Series

Introduction

The STM32CubeF7 firmware package comes with a rich set of examples running on STMicroelectronics boards. The examples are organized by board and provided with preconfigured projects for the main supported toolchains (see *Figure 1*).

Figure 1. STM32CubeF7 firmware components



Reference documents

The reference documents are available on www.st.com/stm32cube:

- Latest release of STM32CubeF7 firmware package
- *Getting started with the STM32CubeF7 firmware package for STM32F7 Series* user manual (UM1891)
- *Description of STM32F7xx HAL drivers* user manual (UM1905)
- *Developing Applications on STM32Cube* with RTOS user manual (UM1722)
- *STM32Cube USB Device library* (UM1734)
- Developing applications on STM32Cube with FatFs (UM1721)

STM32CubeF7 examples

The examples are classified depending on the STM32Cube level they apply to. They are named as follows:

- **Examples:** the examples use only the HAL and BSP drivers (middleware not used). Their objective is to demonstrate the product/peripherals features and usage. They are organized per peripheral (one folder per peripheral, e.g. TIM). Their complexity level ranges from the basic usage of a given peripheral (e.g. PWM generation using timer) to the integration of several peripherals (e.g. how to use DAC for signal generation with synchronization from TIM6 and DMA). The usage of the board resources is reduced to the strict minimum.
- **Applications:** the applications demonstrate the product performance and how to use the available middleware stacks. They are organized either by middleware (a folder per middleware, e.g. USB Host) or by product feature that require high-level firmware bricks (e.g. Audio). The integration of applications that use several middleware stacks is also supported.
- **Demonstrations:** the demonstrations aim to integrate and run the maximum number of peripherals and middleware stacks to showcase the product features and performance.
- **Template project:** the template project is provided to allow to quickly build a firmware application on a given board.

The examples are located under *STM32Cube_FW_STM32CubeF7_VX.Y.Z\Projects*\. They all have the same structure:

- *Inc* folder containing all header files
- *Src* folder containing the sources code
- *EWARM*, *MDK-ARM* and *SW4STM32* folders containing the preconfigured project for each toolchain.
- readme.txt file describing the example behavior and the environment required to run the example.

To run the example, proceed as follows:

1. Open the example using the preferred toolchain.
2. Rebuild all files and load the image into target memory.
3. Run the example by following the readme.txt instructions

Note: Refer to "Development toolchains and compilers" and "Supported devices and evaluation boards" sections of the firmware package release notes to know more about the software/hardware environment used for the firmware development and validation. The correct operation of the provided examples is not guaranteed in other environments, for example when using different compiler or board versions.

The examples can be tailored to run on any compatible hardware: simply update the BSP drivers for your board, provided it has the same hardware functions (LED, LCD display, pushbuttons, etc.). The BSP is based on a modular architecture that can be easily ported to any hardware by implementing the low-level routines.

Table 1 contains the list of examples provided within STM32CubeF7 firmware package.

Table 1. STM32CubeF7 firmware examples

Level	Module Name	Project Name	Description	STM32746G_Discovery	STM32756G-EVAL
Templates	-	Starter project	This directory provides a reference template project that can be used to build any firmware application for STM32F756xx/STM32F746xx devices using STM32CubeF7 HAL and running on STM327x6G-EVAL board from STMicroelectronics.	X	X
Total number of template: 2				1	1
Examples	-	BSP	This example provides a description of how to use the different BSP drivers.	X	X
	ADC	ADC_DualModeInterleaved	This example provides a short description of how to use the ADC peripheral to convert a regular channel in dual interleaved mode.	-	X
		ADC_InjectedConversion_Interrupt	This example describes how to interrupt continuous ADC3 regular ADC_CHANNEL_8 conversion using ADC3 injected ADC_CHANNEL_12 and how to get the converted value of this conversion.	-	X
		ADC_RegularConversion_DMA	This example describes how to use the ADC3 and DMA to transfer continuously converted data from ADC3 to memory.	X	X
		ADC_RegularConversion_Interrupt	This example describes how to use the ADC3 with channel ADC_CHANNEL_8 in interrupt mode to convert data.	-	X
		ADC_RegularConversion_Polling	This example describes how to use the ADC3 with channel ADC_CHANNEL_8 in Polling mode to convert data.	-	X
		ADC_TriggerMode	This example describes how to use the ADC3 and TIM2 to convert continuously data from ADC3 with channel ADC_CHANNEL_8.	-	X
		ADC_TripleModeInterleaved	This example provides a short description of how to use the ADC peripheral to convert a regular channel in triple interleaved mode.	-	X
	CAN	CAN_Loopback	This example provides a description of how to set a communication with the CAN in loopback mode.	-	X
		CAN_Networking	This example shows how to configure the CAN peripheral to send and receive CAN frames in normal mode. The sent frames are used to control LEDs by pressing Tamper pushbutton.	-	X
	CEC	CEC_DataExchange	This example shows how to configure and use the CEC peripheral to receive and transmit messages.	X	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G_Discovery	STM32756G-EVAL
Examples	CRC	CRC_Example	This example guides the user through the different configuration steps by mean of HAL API.	-	X
		CRC_UserDefinedPolynomial	This example guides the user through the different configuration steps by mean of HAL API to ensure the use of the CRC (Cyclic Redundancy Check) calculation unit to get a CRC code of a given buffer of data word (32-bit), based on a user defined generator polynomial. In this example, the polynomial is set manually to 0x9B.	-	X
	CRYP	CRYP_AESModes	This example provides a short description of how to use the CRYPTO peripheral to encrypt and decrypt data using AES in chaining modes (ECB, CBC, CTR) and all key size (128, 192, 256) algorithms.	-	X
		CRYP_AES_CCM	This example provides a short description of how to use the CRYPTO peripheral to encrypt data using AES with Combined Cipher Machine (CCM).	-	X
		CRYP_AES_DMA	This example provides a short description of how to use the CRYPTO peripheral to encrypt and decrypt data using AES-128 algorithm with ECB chaining mode.	-	X
		CRYP_AES_GCM	This example provides a description of how to use the CRYPTO peripheral to encrypt and decrypt data using AES with Galois/Counter Mode (GCM).	-	X
		CRYP_DESTDESmodes	This example provides a short description of how to use the CRYPTO peripheral to encrypt and decrypt data using DES and TDES in all mode (ECB, CBC) algorithm.	-	X
		CRYP_TDES_DMA	This example provides a short description of how to use the CRYPTO peripheral to encrypt data using TDES algorithm.	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G_Discovery	STM32756G-EVAL
Examples	Cortex	CORTEXM_MPU	This example presents the MPU features on STM32F7xx devices and it can be easily ported to any other STM32 device supporting MPU.	-	X
		CORTEXM_ModePrivilege	This example shows how to modify Cortex-M7 thread mode privilege access and stack.	-	X
		CORTEXM_SysTick	This example shows how to use the default configuration of SysTick with a 1 ms time base equal to toggle LEDs.	-	X
	DAC	DAC_SignalsGeneration	This example provides a description of how to use the DAC peripheral to generate several signals using DMA controller.	X	X
		DAC_SimpleConversion	This example provides a short description of how to use the DAC peripheral to do a simple conversion in 8 bits right alignment of 0xFF value, the result of conversion can be seen by connecting PA4 (DAC channel1) to an oscilloscope.	-	X
	DCMI	DCMI_CaptureMode	This example provides a short description of how to use the DCMI to interface with a camera module and to display in continuous mode the picture on the LCD.	-	X
		DCMI_SnapshotMode	This example provides a short description of how to use the DCMI to interface with a camera module and to display in snapshot mode the picture on the LCD.	-	X
	DMA	DMA_FIFOMode	This example provides a description of how to use a DMA stream to transfer a word data buffer from the Flash memory to embedded SRAM memory with FIFO mode enabled through the STM32F7xx HAL API.	-	X
		DMA_FLASHToRAM	This example provides a description of how to use a DMA stream to transfer a word data buffer from the Flash memory to embedded SRAM memory through the HAL API.	X	X
	DMA2D	DMA2D_MemToMemWithBlending	This example provides a description of how to configure DMA2D peripheral in Memory_to_Memory with blending transfer mode.	X	X
		DMA2D_MemToMemWithLCD	This example provides a description of how to configure DMA2D peripheral in Memory_to_Memory transfer mode and display the result on LCD.	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G _Discovery	STM32756G -EVAL
Examples	DMA2D	DMA2D_MemToMemWithPFC	This example provides a description of how to configure DMA2D peripheral for transfer in Memory_to_Memory with Pixel Format Conversion (PFC) mode.	X	X
		DMA2D_MemoryToMemory	This example provides a description of how to configure DMA2D peripheral in Memory_to_Memory transfer mode.	-	X
		DMA2D_RegToMemWithLCD	This example provides a description of how to configure DMA2D peripheral in Register_to_Memory transfer mode and display the result on LCD.	-	X
	FLASH	FLASH_EraseProgram	This example guides the user through the different configuration steps by mean of HAL API how to erase and program the STM32F7xx internal FLASH memory.	X	X
		FLASH_WriteProtection	This example guides the user through the different configuration steps by mean of HAL API how to enable and disable the write protection for the internal FLASH memory integrated within STM32F7xx devices, mounted on STM327x6G-EVAL board revB.	-	X
	FMC	FMC_NOR	This example guides the user through the different configuration steps by mean of HAL API to configure the FMC controller to access the PC28F128M29EWLA NOR memory mounted on STM327x6G-EVAL revB evaluation board.	-	X
		FMC_SDRAM	This example guides the user through the different configuration steps by mean of HAL API to configure the FMC controller to access the IS42S32800G SDRAM memory mounted on STM327x6G-EVAL revB evaluation board.	X	X
		FMC_SDRAM_DataMemory	This example guides the user through the different configuration steps by mean of HAL API to configure the FMC controller to access the IS42S32800G SDRAM memory mounted on STM327x6G-EVAL revB evaluation board (including heap and stack).	-	X
		FMC_SDRAM_LowPower	This example guides the user through the different configuration steps by mean of HAL API to configure the FMC controller to access the IS42S32800G SDRAM memory mounted on STM327x6G-EVAL revB evaluation board, in low-power mode (SDRAM Self Refresh mode).	X	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G_Discovery	STM32756G-EVAL
Examples	FMC	FMC_SDRAM_MemRemap	This example guides the user through the different configuration steps to use the IS42S32800G SDRAM memory (mounted on STM327x6G-EVAL revB evaluation board) as code execution memory.	-	X
		FMC_SRAM	This example guides the user through the different configuration steps by mean of HAL API to configure the FMC controller to access the IS61WV102416BLL-10MLI SRAM memory mounted on STM327x6G-EVAL revB evaluation board.	-	X
		FMC_SRAM_DataMemory	This example guides the user through the different configuration steps by mean of HAL API to configure the FMC controller to access the IS61WV102416BLL-10MLI SRAM memory mounted on STM327x6G-EVAL revB evaluation board (including heap and stack).	-	X
	GPIO	GPIO_EXTI	This example shows how to configure external interrupt lines.	-	X
		GPIO_IOToggle	This example describes how to configure and use GPIOs through the STM32F7xx HAL API.	-	X
	HAL	HAL_TimeBase	This example describes how to customize the HAL time base using a general purpose timer (TIM6) instead of Systick as main source of time base.	-	X
	HASH	HASH_HMAC_SHA1MD5	This example provides a short description of how to use the HASH peripheral to hash data using HMAC SHA-1 and HMAC MD5 algorithms.	-	X
		HASH_SHA1MD5	This example provides a short description of how to use the HASH peripheral to hash data using SHA-1 and MD5 algorithms.	-	X
		HASH_SHA1MD5_DMA	This example provides a short description of how to use the HASH peripheral to hash data using SHA-1 and MD5 algorithms.	-	X
		HASH_SHA224SHA256_DMA	This example provides a short description of how to use the HASH peripheral to hash data using SHA224 and SHA256 algorithms.	-	X
	I2C	I2C_EEPROM	This example describes how to perform I2C data buffer transmission/reception via DMA.	-	X
		I2C_TwoBoards_AdvComIT	This example describes how to perform I2C data buffer transmission/reception between two boards, using an interrupt.	X	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G_Discovery	STM32756G-EVAL
Examples	I2C	I2C_TwoBoards_ComDMA	This example describes how to perform I2C data buffer transmission/reception between two boards, via DMA.	X	-
		I2C_TwoBoards_ComIT	This example describes how to perform I2C data buffer transmission/reception between two boards using an interrupt.	X	-
		I2C_TwoBoards_ComPolling	This example describes how to perform I2C data buffer transmission/reception between two boards in Polling mode.	X	-
	IWDG	IWDG_Example	This example describes how to reload the IWDG counter and to simulate a software fault by generating an MCU IWDG reset when a programmed time period has elapsed.	-	X
	LTDC	LTDC_ColorKeying	This example describes how to enable and use the LTDC color keying functionality.	-	X
		LTDC_Display_1Layer	This example provides a description of how to configure LTDC peripheral to display BMP image of size 480x272 and format RGB888 (24 bits/pixel) on LCD using only one layer.	X	X
		LTDC_Display_2Layers	This example describes how to configure the LTDC peripheral to display two layers at the same time.	X	X
	PWR	PWR_CurrentConsumption	This example shows how to configure the STM32F7xx system to measure different Low-power mode current consumption. The Low-power modes are: - Sleep mode - Stop mode with RTC - Standby mode without RTC and BKPSRAM - Standby mode with RTC - Standby mode with RTC and BKPSRAM To run this example, the user has to follow the following steps: 1. Select the Low-power modes to be measured by uncommenting the corresponding line inside the stm32f7xx_lp_modes.h file.	X	-
		PWR_STANDBY	This example shows how to enter the system to Standby mode and wake-up from this mode using: external RESET, RTC Alarm A or WKUP pin.	-	X
		PWR_STOP	This example shows how to enter the system to Stop mode and wake-up from this mode using RTC Wakeup Timer Event connected to EXTI_Line22 or Tamper pushbutton EXTI15_10.	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G_Discovery	STM32756G-EVAL
Examples	QSPI	QSPI_ExecuteInPlace	This example describes how to configure and use QPSI through the STM32F7xx HAL API.	X	X
		QSPI_MemoryMapped	This example describes how to configure and use QPSI through the STM32F7xx HAL API.	-	X
		QSPI_ReadWrite_DMA	This example describes how to configure and use QPSI through the STM32F7xx HAL API.	-	X
		QSPI_ReadWrite_IT	This example describes how to configure and use QPSI through the STM32F7xx HAL API.	X	X
	RCC	RCC_ClockConfig	This example describes how to use the RCC HAL API to configure the system clock (SYSCLK) and modify the clock settings in run mode.	X	X
	RNG	RNG_MultiRNG	This example guides the user through the different configuration steps by mean of HAL API to ensure RNG random 32bit number generation.	-	X
	RTC	RTC_Alarm	This example guides the user through the different configuration steps by mean of HAL API to configure and generate an RTC alarm.	-	X
		RTC_Calendar	This example guides the user through the different configuration steps by mean of HAL API to ensure Calendar configuration using the RTC peripheral.	-	X
		RTC_Tamper	This example guides the user through the different configuration steps by mean of HAL API to write/read data to/from RTC Backup registers and demonstrate the Tamper detection feature.	-	X
		RTC_TimeStamp	This example guides the user through the different configuration steps by mean of HAL API to ensure Time Stamp configuration using the RTC peripheral.	-	X
	SAI	SAI_Audio	This example provides basic implementation of audio features using BSP_AUDIO.	-	X
		SAI_AudioPlay	This example shows how to play an audio file using the DMA circular mode and how to handle the buffer update.	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G_Discovery	STM32756G-EVAL
Examples	SPI	SPI_FullDuplex_ComDMA	This example shows how to perform SPI data buffer transmission/reception between two boards via DMA.	X	-
		SPI_FullDuplex_ComIT	This example shows how to ensure SPI data buffer transmission/reception between two boards by using an interrupt.	X	-
		SPI_FullDuplex_ComPolling	This example shows how to ensure SPI data buffer transmission/reception in Polling mode between two boards.	X	-
	TIM	TIM_6Steps	This example shows how to configure the TIM1 peripheral to generate 6 Steps.	-	X
		TIM_7PWMOutput	This example shows how to configure the TIM1 peripheral to generate 7 PWM signals with 4 different duty cycles (50%, 37.5%, 25% and 12.5%).	-	X
		TIM_CascadeSynchro	This example shows how to synchronize TIM2 and Timers (TIM3 and TIM4) in cascade mode.	-	X
		TIM_ComplementarySignals	This example shows how to configure the TIM1 peripheral to generate three complementary TIM1 signals, to insert a defined dead time value, to use the break feature and to lock the desired parameters.	-	X
		TIM_DMA	This example provides a description of how to use DMA with TIM2 update request to transfer Data from memory to TIM2 Capture Compare Register 3 (CCR3).	-	X
		TIM_DMABurst	This example shows how to update the TIM2 channel1 period and the duty cycle using the TIM2 DMA burst feature.	-	X
		TIM_ExtTriggerSynchro	This example shows how to synchronize TIM peripherals in cascade mode with an external trigger.	-	X
		TIM_InputCapture	This example shows how to use the TIM peripheral to measure the frequency of an external signal.	-	X
		TIM_OCActive	This example shows how to configure the TIM peripheral in Output Compare Active mode (when the counter matches the capture/compare register, the concerned output pin is set to its active state).	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G _Discovery	STM32756G -EVAL
Examples	TIM	TIM_OCInactive	This example shows how to configure the TIM peripheral in Output Compare Inactive mode with the corresponding Interrupt requests for each channel.	-	X
		TIM_OCToggle	This example shows how to configure the TIM peripheral to generate four different signals with four different frequencies.	-	X
		TIM_OnePulse	This example shows how to use the TIM peripheral to generate a One pulse mode after a Rising edge of an external signal is received in Timer Input pin.	-	X
		TIM_PWMInput	This example shows how to use the TIM peripheral to measure the frequency and duty cycle of an external signal.	-	X
		TIM_PWMOutput	This example shows how to configure the TIM peripheral in PWM (Pulse Width Modulation) mode.	-	X
		TIM_ParallelSynchro	This example shows how to synchronize TIM2 and timers (TIM3 and TIM4) in parallel mode.	-	X
		TIM_PrescalerSelection	This example shows how to configure the TIM peripheral in PWM (Pulse Width Modulation) mode with clock prescaler selection feature activated using __HAL_RCC_TIMCLKPRESCALER() which allow to double the output frequency.	-	X
		TIM_Synchronization	This example shows how to synchronize TIM1 and Timers (TIM3 and TIM4) in parallel mode.	-	X
		TIM_TimeBase	This example shows how to configure the TIM peripheral to generate a time base of one second with the corresponding Interrupt request.	X	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G_Discovery	STM32756G-EVAL
Examples	UART	UART_HyperTerminal_DMA	This example guides the user through the different configuration steps by mean of HAL API to ensure UART Data buffer transmission and reception with DMA.	-	X
		UART_HyperTerminal_IT	This example guides the user through the different configuration steps by mean of HAL API to ensure UART Data buffer transmission and reception with interrupt.	-	X
		UART_Printf	This example shows how to retarget the C library printf function to the UART.	-	X
		UART_TwoBoards_ComDMA	This example describes a UART transmission (transmit/receive) in DMA mode between two boards.	X	-
		UART_TwoBoards_ComIT	This example describes a UART transmission (transmit/receive) in interrupt mode between two boards.	X	-
		UART_TwoBoards_ComPolling	This example describes a UART transmission (transmit/receive) in polling mode between two boards.	X	-
	WWDG	WWDG_Example	This example guides the user through the different configuration steps by mean of HAL API to ensure WWDG counter update and simulate a software fault that generates an MCU WWDG reset when a predefined time period has elapsed.	-	X
Total number of examples: 117				27	90
Demonstration	-	-	The STM32Cube demonstration platform comes on top of the STM32Cube as a firmware package that offers a full set of software components based on a module architecture allowing re-using them separately in standalone applications. All these modules are managed by the STM32Cube demonstration kernel allowing to dynamically adding new modules and access to common resources (storage, graphical components and widgets, memory management, Real-Time operating system). The STM32Cube demonstration platform is built around the powerful graphical library STemWin and the FreeRTOS real time operating system and uses almost the whole STM32 capability to offer a large scope of usage based on the STM32Cube HAL BSP and several middleware components.	X	X
Total number of demonstrations: 2				1	1

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G_Discovery	STM32756G-EVAL
Applications	Audio	Audio_playback_and_record	This application shows how to use the different functionalities of audio device and ST MEMS microphones (MP45DT02), three different menus are available to switch between them use the joystick button: Explorer Audio File menu, Start Audio Player menu and Start Audio Recorder menu.	X	X
	Camera	Camera_To_USBDisk	This application provides a short description of how to use the DCMI to interface with the camera module and display in continuous mode the picture on LCD and to save a picture in USB device.	-	X
	Display	LTDC_AnimatedPictureFromSDCard	This application describes how to display an animated picture on LCD saved under microSD.	-	X
		LTDC_Paint	This application describes how to configure LCD touch screen and attributes an action related to configured touch zone and how to save BMP picture in USB Disk.	-	X
		LTDC_PicturesFromSDCard	This application describes how to display pictures on LCD saved under SD card.	X	X
	FatFs	FatFs_MultiDrives	This application provides a description on how to use STM32Cube firmware with FatFs middleware component as a generic FAT file system module, in order to develop an application exploiting FatFs offered features with multidrives (RAMDisk, uSD) configuration.	X	X
		FatFs_RAMDisk	This example provides a description on how to use STM32Cube firmware with FatFs middleware component as a generic FAT file system module, in order to develop an application exploiting FatFs offered features with RAM disk (SDRAM) drive configuration.	-	X
		FatFs_RAMDisk_RTOS	This application provides a description on how to use STM32Cube firmware with FatFs middleware component as a generic FAT file system module, in order to develop an application exploiting FatFs offered features with RAM disk (SDRAM) drive in RTOS mode configuration.	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G _Discovery	STM32756G -EVAL
Applications	FatFs	FatFs_USBDisk	This application provides a description on how to use STM32Cube firmware with FatFs middleware component as a generic FAT file system module and STM32 USB On-The-Go (OTG) host library, in Full Speed (FS), High Speed (HS) and High Speed in Full Speed (HS-IN-FS) modes, in order to develop an application exploiting FatFs offered features with USB disk drive configuration.	-	X
		FatFs_USBDisk_MultipleAccess_ RTOS	This application provides a description on how to use STM32Cube firmware with FatFs middleware component as a generic FAT file system module, FreeRTOS as an RTOS module based on using CMSIS-OS wrapping layer common APIs, and also STM32 USB On-The-Go (OTG) host library, in Full Speed (FS), High Speed (HS) and High Speed in Full Speed (HS-IN-FS) modes, in order to develop an application exploiting FatFs offered features with USB disk drive in RTOS mode configuration.	-	X
		FatFs_USBDisk_RTOS	This application provides a description on how to use STM32Cube firmware with FatFs middleware component as a generic FAT file system module, FreeRTOS as an RTOS module based on using CMSIS-OS wrapping layer common APIs, and also STM32 USB On-The-Go (OTG) host library, in Full Speed (FS), High Speed (HS) and High Speed in Full Speed (HS-IN-FS) modes, in order to develop an application exploiting FatFs offered features with USB disk drive in RTOS mode configuration.	-	X
		FatFs_uSD	This example provides a description on how to use STM32Cube firmware with FatFs middleware component as a generic FAT file system module. The objective is to develop an application making the most of the features offered by FatFs to configure a microSD drive.	X	X
		FatFs_uSD_RTOS	This application provides a description on how to use STM32Cube firmware with FatFs middleware component as a generic FAT file system module, in order to develop an application exploiting FatFs offered features with microSD drive in RTOS mode configuration.	X	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G_Discovery	STM32756G-EVAL
Applications	Free RTOS	FreeRTOS_DelayUntil	This directory contains a set of source files that implement thread delaying using osDelayUntil function.	X	X
		FreeRTOS_LowPower	This directory contains a set of source files that implement an application that uses message queues with CMSIS RTOS API. This application creates two threads.	-	X
		FreeRTOS_Mail	This directory contains a set of source files that implement an example that uses mail queues with CMSIS RTOS API. This example creates two threads that send and receive mail. The mail to send/receive is a structure that holds three variables (var1 and var2 are uint32, var3 is a uint8). One thread acts as a producer and the other as the consumer.	X	X
		FreeRTOS_Mutexes	This directory contains a set of source files that implement an application that uses mutexes with CMSIS RTOS API This application creates three threads with different priorities, and access the same mutex MutexHighPriorityThread() has the highest priority so executes first and grabs the mutex and sleeps for a short period to let the lower priority threads execute. When it has completed its demo functionality it gives the mutex back before suspending itself.	-	X
		FreeRTOS_Queues	This directory contains a set of source files that implement an application that uses message queues with CMSIS RTOS API. This application creates two threads that send and receive an incrementing number to/from a queue.	-	X
		FreeRTOS_Semaphore	This directory contains a set of source files that implement an application that uses semaphores with CMSIS RTOS API. This application creates two threads that toggle LEDs through a shared semaphore.	-	X
		FreeRTOS_SemaphoreFromISR	This directory contains a set of source files that implement an application that uses semaphore from ISR with CMSIS RTOS API. This application creates a thread that toggles LED through semaphore given from ISR.	X	X
		FreeRTOS_Signal	This directory contains a set of source files that implement thread signaling example using CMSIS RTOS API.	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G_Discovery	STM32756G-EVAL
Applications	Free RTOS	FreeRTOS_SignalFromISR	This directory contains a set of source files that implement a thread signalling from an interrupt example using CMIS RTOS API.	-	X
		FreeRTOS_ThreadCreation	This directory contains a set of source files that implement a thread creation example using CMSIS RTOS API. This example creates two threads with the same priority, which execute in a periodic cycle of 15 seconds.	-	X
		FreeRTOS_Timers	This directory contains a set of source files that implement an application that uses timers of CMSIS RTOS API. This application creates a thread that toggles LED2 every 400 ms, and a periodic timer that calls a callback function every 200 ms to toggle the LED1.	-	X
	IAP	IAP_Binary_Template	This directory contains a set of sources files that build the application to be loaded into Flash memory using In-Application Programming (IAP, through USART).	-	X
		IAP_Main	This directory contains a set of source files and pre-configured projects that describes how to build an application to be loaded into the Flash memory using In-Application Programming (IAP, through USART).	-	X
	LibJPEG	LibJPEG_Decoding	This application demonstrates how to read jpeg file from SD card memory, decode it and display the final BMP image on the LCD.	X	X
		LibJPEG_Encoding	This example demonstrates how to read BMP file from micro SD, encode it, save the jpeg file in the SD card memory, then decode the jpeg file and display the final BMP image on the LCD.	-	X
	LwIP	LwIP_HTTP_Server_Netconn_RTOS	This application guides STM32Cube HAL API users to run a http server application based on Netconn API of LwIP TCP/IP stack. The communication is done with a web browser application in a remote PC.	X	X
		LwIP_HTTP_Server_Raw	This application guides STM32Cube HAL API users to run a http server application based on Raw API of LwIP TCP/IP stack The communication is done with a web browser application in a remote PC.	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G_Discovery	STM32756G-EVAL
Applications	LwIP	LwIP_HTTP_Server_Socket_RTOS	This application guides STM32Cube HAL API users to run a http server application based on Socket API of LwIP TCP/IP stack The communication is done with a web browser application in a remote PC.	-	X
		LwIP_IAP	This application guides STM32Cube HAL API users to run In-Application Programming (IAP) over Ethernet.	-	X
		LwIP_TCP_Echo_Client	This application guides STM32Cube HAL API users to run TCP Echo Client application based on Raw API of LwIP TCP/IP stack. To run this application, on the remote PC, open a command prompt window.	-	X
		LwIP_TCP_Echo_Server	This application guides STM32Cube HAL API users to run TCP Echo Server application based on Raw API of LwIP TCP/IP stack. To run this application, on the remote PC, open a command prompt window.	-	X
		LwIP_TFTP_Server	This application guides STM32Cube HAL API users to run a TFTP server application for STM32F7xx devices.	-	X
		LwIP_UDPTCP_Echo_Server_Netconn_RTOS	This application guides STM32Cube HAL API users to run a UDP/TCP Echo Server application based on Netconn API of LwIP TCP/IP stack. To run this application, on the remote PC, open a command prompt window.	-	X
		LwIP_UDP_Echo_Client	This application guides STM32Cube HAL API users to run a UDP Echo Client application based on Raw API of LwIP TCP/IP stack. To run this application, on the remote PC, open a command prompt window.	-	X
		LwIP_UDP_Echo_Server	This application guides STM32Cube HAL API users to run UDP Echo Server application based on Raw API of LwIP TCP/IP stack. To run this application, on the remote PC, open a command prompt window.	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G_Discovery	STM32756G-EVAL
Applications	PolarSSL	SSL_Client	This application guides STM32Cube HAL API users to run an SSL client application based on PolarSSL crypto library and LwIP TCP/IP stack to off-load the CPU from encryption/decryption, hash and RNG, all these algorithms are implemented using the hardware acceleration AES 128/192/256, Triple DES, MD5, SHA-1 and analog RNG through the STM32Cube HAL APIs In this application the client (STM32756G-EVAL) sends a crypted message to the server (test PC), which will decrypt the message then reply to the client.	-	X
		SSL_Server	This application guides STM32Cube HAL API users to run an SSL Server application based on PolarSSL crypto library and LwIP TCP/IP stack to off-load the CPU from encryption/decryption, hash and RNG, all these algorithms are implemented using the hardware acceleration AES 128/192/256, Triple DES, MD5, SHA-1, SHA2-2 and analog RNG through the STM32Cube HAL APIs the HTTP server (STM32756G-EVAL) contains a html page dynamically refreshed (every 1 s), it shows the RTOS statistics in runtime The HyperTerminal can be used to debug messages exchanged between the client and server.	-	X
	QSPI	QSPI_perfs	This application describes how to display pictures stored on QSPI flash memory on LCD and measures data transfer performance between QSPI flash and SDRAM memory.	X	-
	STemWin	STemWin_HelloWorld	This directory contains a set of source files that implement a simple "Hello World" application based on STemWin for STM32F7x6 devices.	X	X
		STemWin_SampleDemo	This directory contains a set of source files that implement a sample demonstration application allowing to show some of the STemWin Library capabilities on STM32F7x6 devices.	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G_Discovery	STM32756G-EVAL
Applications	USB_ Device	AUDIO_Standalone	This application is a part of the USB Device Library package using STM32Cube firmware. It describes how to use USB device application based on the AUDIO Class implementation of an audio streaming (Out: Speaker/Headset) capability on the STM32F7xx devices.	X	X
		CDC_Standalone	This application shows how to use the USB device application based on the Device Communication Class (CDC) compliant with the PSTN subprotocol. The USB Device and UART peripherals are used.	-	X
		CustomHID_Standalone	This application is a part of the USB Device Library package using STM32Cube firmware. It describes how to use USB device application based on the Custom HID Class on the STM32F7xx devices.	-	X
		DFU_Standalone	This application is a part of the USB Device Library package using STM32Cube firmware. It describes how to use USB device application based on the Device Firmware Upgrade (DFU) on the STM32F7xx devices.	X	X
		DualCore_Standalone	This application is a part of the USB Device Library package using STM32Cube firmware. It describes how to use USB device application based on the STM32F7x6 multi core support feature integrating the Device Communication Class (CDC) and Human Interface (HID) in the same project.	X	X
		HID_LPM_Standalone	The STM32F7x6 devices support the USB Link Power Management Protocol (LPM-L1) and complies with the USB 2.0 LPM-L1 ECN. The hpcd.Init.lpm_enable in the usbd_conf.c should be set to 1 to enable the support for LPM-L1 protocol in the USB stack.	X	X
		HID_Standalone	This application is a part of the USB Device Library package using STM32Cube firmware. It describes how to use USB device application based on the Human Interface (HID) on the STM32F7x6 devices.	X	X
		MSC_Standalone	This application is a part of the USB Device Library package using STM32Cube firmware. It describes how to use USB device application based on the Mass Storage Class (MSC) on the STM32F7x6 devices.	X	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G_Discovery	STM32756G-EVAL
Applications	USB_ Host	AUDIO_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the Audio OUT class on the STM32F7xx devices.	-	X
		CDC_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the Communication Class (CDC) on the STM32F7xx devices.	X	X
		DualCore_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the STM32F7x6 multi core support feature integrating Mass Storage (MSC) and Human Interface (HID) in the same project.	X	X
		DynamicSwitch_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use dynamically switch, on the same port, between available USB host applications on the STM32F7x6 devices.	X	X
		FWupgrade_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the In-Application programming (IAP) on the STM32F7x6 devices.	-	X
		HID_RTOS	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the Human Interface Class (HID) on the STM32F7x6 devices.	X	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32746G _Discovery	STM32756G -EVAL
Applications	USB_ Host	HID_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the Human Interface Class (HID) on the STM32F7x6 devices.	X	X
		MSC_RTOS	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the Mass Storage Class (MSC) on the STM32F7x6 devices in RTOS mode configuration.	X	X
		MSC_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the Mass Storage Class (MSC) on the STM32F7x6 devices.	X	X
		MTP_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the Media Transfer Protocol (MTP) on the STM32F7x6 devices.	-	X
Total number of applications: 86				25	61
Total number of projects: 207				54	153

1 Revision history

Table 2. Document revision history

Date	Revision	Changes
07-Jul-2015	1	Initial release.